

SYSTEM AND METHOD FOR SHARING RESOURCE

Patent number: JP10049390
Publication date: 1998-02-20
Inventor: MILITO RODOLFO A
Applicant: AT & T CORP
Classification:
- international: G06F9/46; H04L12/28; H04L12/56; H04M3/00; H04Q3/00; H04Q3/545
- european:
Application number: JP19960290975 19961101
Priority number(s):

Also published as:

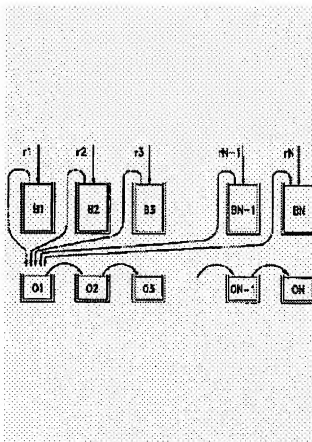


EP0772324 (A)
US5596576 (A)
EP0772324 (A)

Abstract of JP10049390

PROBLEM TO BE SOLVED: To assign an access to the resource that is limited and shared by plural users having access requests to the resource by permitting the accesses to the resource as long as a prescribed number of tokens are included in a related bank.

SOLUTION: The bank Bi of every class receives a token stream at a contracted rate r_i , and the overflow bank Oi is related with every class. If a bank Bi that is filled with the tokens generated to a specific class is detected, these tokens are not immediately put into the bank Oi of the relevant class. Then the extra tokens are to be put into a bank O1 of higher priority. If the bank O1 is filled with tokens, the extra tokens are put into an overflow bank O2. Then the accesses are permitted to the resource as long as a prescribed number of tokens are included in a relate bank. Otherwise, the accesses are rejected to the resource.



BEST AVAILABLE COPY

特開平10-49390

(43) 公開日 平成10年(1998) 2月20日

(51) Int.Cl. ⁴	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 9/46	3 4 0		G 0 6 F 9/46	3 4 0 F
H 0 4 L 12/28			H 0 4 M 3/00	D
	12/56		H 0 4 Q 3/00	
H 0 4 M 3/00			3/545	
H 0 4 Q 3/00			H 0 4 L 11/00	3 1 0 D

審査請求 未請求 請求項の数16 OL (全 14 頁) 最終頁に続く

(21) 出願番号 特願平8-290975

(22) 出願日 平成8年(1996)11月1日

(31) 優先権主張番号 08/552899

(32) 優先日 1995年11月3日

(33) 優先権主張国 米国 (US)

(71) 出願人 390035493

エイ・ティ・アンド・ティ・コーポレーション
 AT&T CORP.
 アメリカ合衆国 10013-2412 ニューヨ
 ーク ニューヨーク アヴェニュー オフ
 ジ アメリカズ 32

(72) 発明者 ルドルフォ エー. ミルト
 アメリカ合衆国 08854 ニュージャーシ
 イ. ビスカタウェイ, ザークル アヴェニ
 ュー 9

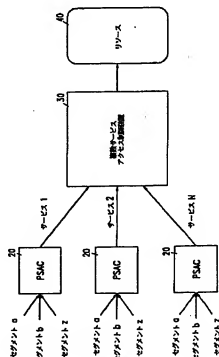
(74) 代理人 弁理士 岡部 正夫 (外1名)

(54) 【発明の名称】 リソース共用のためのシステムおよび方法

(57) 【要約】 (修正有)

【課題】 制限されたリソースを多数の使用者および／または使用者クラスの間で共用するためのシステムおよび方法に関する。

【解決手段】 各使用者に関係付けられた最小限の保証されたレートによって決定されるレートでトークンを各使用者に割り当て、一方各使用者に関係付けられたバンクに各使用者に割り当てられたトークンを蓄積して、各バンクは各使用者に関係付けられた有限の容量を持っている。その関係付けられたバンクの容量に到達した使用者に割り当てられたトークンを、そのバンクの容量が小さい他の使用者のバンクに蓄積する。使用者があらかじめ決められた数のトークンをその関係付けられたバンクに備えている場合は、リソースにアクセスを要求している使用者にそのリソースへのアクセスをさせて、さもなければリソースへのアクセスを拒絶する。



【特許請求の範囲】

【請求項1】 共用される限定されたリソースにアクセスを要求する複数の使用者の間で前記リソースにアクセスを割り当てるための方法であって、前記割り当てを、各使用者と関係付けられた最小限の保証されたレートでの前記リソースへのアクセスを各使用者が許可されるように成して、そして前記リソースが利用できる場合だけはその最小限の保証されたレートを越えているいかなる使用者によっても前記リソースが利用できるように成して、前記方法は、

各使用者に関係付けられた最小限の保証されたレートによって決定されるレートで各使用者にトークンを割り当て、前記方法は、

各使用者に割り当てられたトークンを各使用者に関係付けられたバンクに蓄積するステップと、ここで前記各バンクは各使用者に関係付けられた有限の容量を備えている、

その関係付けられたバンクがその容量に到達した使用者に割り当てられたトークンを、そのバンクがその容量に到達していない別の使用者に割り当てられたバンクに蓄積するステップと、そして、

前記使用者がその関係付けられたバンクにあらかじめ決められた数のトークンを持っている場合は、前記リソースにアクセスを要求している使用者に前記リソースへアクセスすることを許可し、そうでない場合は、前記リソースへのアクセスを拒絶するステップと、

から構成されることを特徴とする方法。

【請求項2】 請求項1記載の方法において、前記使用者が別々の優先権を割り当てられていて、前記別の使用者は前記割り当てられた優先権に基づいて選択されることを特徴とする方法。

【請求項3】 請求項1記載の方法においてさらに、前記リソースにアクセスを許可された使用者に関係付けられたバンクからあらかじめ決められた数のトークンを削除するステップから構成されることを特徴とする方法。

【請求項4】 請求項1記載の方法において、前記あらかじめ決められたトークンの数が1であることを特徴とする方法。

【請求項5】 請求項1記載の方法において、前記バンクの各々が、あらかじめ決められた同一の有限の容量を備えていることを特徴とする方法。

【請求項6】 請求項1記載の方法において、前記バンクの少なくとも二つが異なるあらかじめ決められた有限の容量を備えていることを特徴とする方法。

【請求項7】 共用されたリソースへのアクセスを割り当てるための方法であって、(a)前記リソースの複数の使用者の各々に対して加算できるカウンタを確立するステップと、この前記加算できるカウンタは計数を持っていて、(b)前記計数が各使用者に関係付けられてあらかじめ決められた最大計数値より小さい場合には、各使

用者に対して確立された各カウンタの計数を各使用者に関係付けられたレートで増加させるステップと、(c)

アクセスを要求している前記使用者に対して確立されたカウンタの計数がゼロでない場合には、前記リソースにアクセスを要求している使用者にアクセスすることを許可するステップと、(d)ステップ(b)で増加させられた他のどのカウンタの計数もそのあらかじめ定められた最大計数にある場合は、そのカウンタが前記あらかじめ決められた最大計数より小さい使用者に関係付けられたカウンタの計数をその関係付けられたレートよりも高いレートで増加させるステップと、から構成されることを特徴とする方法。

【請求項8】 請求項7記載の方法において、前記使用者が異なる優先権を割り当てられていて、ステップ(d)で選択された前記使用者が前記割り当てられた優先権に基づいて選択されることを特徴とする方法。

【請求項9】 請求項7記載の方法においてさらに、アクセスを要求している前記使用者が前記リソースへのアクセスを許可された場合は、アクセスを要求している使用者に対して確立されたカウンタの計数を削減するステップから構成されることを特徴とする方法。

【請求項10】 請求項7記載の方法において、前記各々の使用者が、その関係付けられたあらかじめ決められた同一の最大計数を備えていることを特徴とする方法。

【請求項11】 請求項7記載の方法において、前記使用者の少なくとも二つがあらかじめ決められた異なる最大計数を備えていることを特徴とする方法。

【請求項12】 N人の使用者の間での共有リソースへのアクセスを共用する方法であって、ここでNは1より大きな整数であり、

前記N人の使用者の各々にレート「R_i」を割り当てるステップと、ここで、前記レートは1番目の使用者に対して最小限必要なアクセスのレートを示しており、 $1 \leq i \leq N$ 、

i番目の使用者の前記共用リソースにアクセスする資格のしるし「B_i」を前記N人の使用者の各々に対して蓄積するステップと、

前記N個の使用者の各々に割り当てられたあらかじめ決められた限度値L_iまで、その関係付けられたレートR_iで各B_iを増加するステップと、

他のどのB_iもその関係付けられた限度値L_iにあって、かつ前記選択されたB_iがその限度値L_iにない場合は、その関係付けられたR_iよりも大きなレートで、選択されたB_iを増加させるステップと、そして、

その関係付けられたB_iがゼロでない場合は、その使用者に前記共有リソースへアクセスすることを許可するステップと、

から構成されることを特徴とする方法。

【請求項13】 請求項12記載の方法においてさら

3

に、前記リソースの1番目の使用者にアクセスを許可する際に、B_iを削減するステップから構成されることを特徴とする方法。

【請求項14】 請求項12記載の方法において、前記使用者が異なる優先権を割り当てられて、前記選択されたB_iが前記割り当てられた優先権に基づいて選択されることを特徴とする方法。

【請求項15】 請求項12記載の方法において、前記使用者の各々に関係付けられた限度値L_iが同じ値を備えていることを特徴とする方法。

【請求項16】 請求項12記載の方法において、前記使用者に関係付けられた少なくとも2つの限度値L_iが異なる値であることを特徴とする方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、制限されたリソースを多数の使用者および/または使用者クラスの間で共用するためのシステムおよび方法に関する。

【0002】

【従来の技術】 リソースは多数の使用者によって共用されることが多い。たとえば、電気通信サービスの供給者は、顧客の要求にサービスするために重要なリソースをある程度だけ数だけしか備えていない。使用されていないリソースを充分に利用させる一方で重要な使用者にシステムの性能低下を起こさせないようにするために、リソースを割り当てるための公平で効率的なシステムが望まれている。

【0003】 共用されたリソース環境の中でリソースの公平な共用を保証するという議論は現在のビジネス環境に広がっている。共用は原理的に規模の利益に到達することができること、また需要の非同時性（たとえば、ビジネス地域対住宅地域のトラフィック）が活用できることが広く認識されている。他の顧客からの予想も無い大きな需要に直面しても使用者の契約したリソースの「持ち分」に対してリアルタイムのアクセスを保証するような、完全なフレーム・ワークは存在しない。リソースの完全な分割によって保護を達成することができる。そのような分割は、使用者がお互いから保護されている場合には効果的だが、リソースが特定の使用者に割り当てられていても使用されていない場合は無駄になるので効率的ではない。

【0004】

【発明が解決しようとする課題】 典型的なリソースの共用状況においては、通信およびコンピュータのネットワークによって例証されるように、三つの大きな課題について言及する必要がある。すなわち、システムまたはサブシステムへのアクセスを制御する課題、特定のプロセッサを予定に入れる課題、および個別のリソースを割り当てる課題である。

【0005】 課題と解決策を視覚化できるようにするた

4

めに、AT&T社の5ESS[®] 交換機を参照する。図1は、ストリーム₁からストリーム₂で表される複数のクラスのトラフィックの要求に従う5ESS[®]を図示する。典型的に、ストリームは特定のサービスの要求を束ねており、たとえば呼び出し設定を含むことができる。たとえば、無線サービスに関連したストリームは、ハンドオーバ要求、位置アップデート、その他を含むことができる。一定の活動が、リンク、プロセッサ、データベース、トラunkなどのような多数のリソースに要求を配置する。

【0006】 5ESS[®]は、ある種の持続されたトラフィック混合に従うあるグレードのサービスを提供することができるように設計され設備されている。基本的なアクセス制御機構は、どのクラスもその契約の値を越えないことを保証するように実行されることになっている。そのようなアプローチは効果的ではあるが、効率的とは言いがたい。結局、他のクラスがその契約のレート以下にあるのでシステムが容量以下で動作しているとき、その割り当てられたレートを越えるクラスへのアクセスは拒絶されてはならない。トラフィックは可変であるのでこれは普通に発生する。一つのクラスがその契約のレートを越えることはありえるが、すべてのクラスがその最大の持ち分を要求しているときにはこれは必ずしも発生しない。

【0007】 第二の課題、すなわちプロセッサの共用は本来ローカルなものである。包括的なアクセス制御に加えてプロセッサの共用を制御することが必要なことがある。たとえば、特定のプロセッサが包括的なリソースとして重要な役割を演ずることがある。第二に、異なるクラスはいろいろなリソースに種々の負担を課す。まさしくその性質によって、包括的なアクセス機構は、あらゆるリソースの各々のクラスの要求の評価に基づいていなければならない。さらに、単一のクラスそれ自体が多数の活動を含むことがある。したがって、この評価は活動のある種の混合を仮定する。しかし、いったんあるクラスがアクセスを得たら、それが将来もそのような混合を持つという保証はない。

【0008】 最後に、個別のリソースの共用の課題に直面する。一例は、5ESS[®]の中のトラunkに対する競合である。

【0009】 アメリカ特許第5,274,644号（「644特許」：参考のためここで組み入れた）は、共通リソースへのアクセスを制御するための方法のいくつかの事例を開示しており、共通リソースへのアクセスを制御するために「トークン」を利用する機構を含んでいる。「644特許」のシステムは、リソースへの使用者のアクセスがトークンへの使用者のアクセスに基づくように統制する、各使用者に関係付けられた「スロットル」を含んでおり、これを以下に説明する。

【0010】 図2は、単一ストリームに対するレート制

御スロットルを図示する。本システムは、トラフィックの重要なパラメータ（到着率、ピーク）がある限度内に保たれるかぎり、望ましいグレードのサービスを提供することができる。レート制御スロットルは、長い目で見れば、システムへの入場許可レートがあらかじめ決められた値を越えないことを保証して、そのピークを制限することによってトラフィックを整えている。このあらかじめ決められた値が、各使用者に対する契約の入場許可レート r_i であり ($i = 1 \sim N$, N は使用者の合計数である)、およびピークに関連した因子 L_i を確立するために使用される。この機構は、各使用者に対する、有限の容量 L_i のバンク B_i と、あらかじめ決められたレート r_i でトークンを生成する発信源 (図示せず) とから構成される。入場許可機構は、次のように動作する。

- 1) トークンは、レート r_i で生成される。
- 2) すでにバンクの中にあるトークンの数が L_i より少ない場合は、新たに生成されたトークンはバンク B_i の中に挿入される。そうでない場合は、トークンは破棄される。
- 3) バンク B_i の中にトークンを発見するユーザー i からの到着は、サービスに入るのを許可される (そして、そのトークンは破棄される)。そうでない場合は、その到着は拒絶されるかまたは待ち行列に入れられる。

【0011】上に示したように、レート制御スロットルは、各クラスに対して別々のバンクを提供することにより N 個のクラスの使用者の事例に一般化することができるが、完全に効率的ではない。いくつかのクラスが一定のリソースを求めて競合するとき、その内のいくつかはその契約レベル以下で動作することがありそうである。バンクが満杯であることを見つけたトークンが破棄されるレートが、リソースの利用不足の指標である。(もちろんリソースの提供とトークン・レートの割り当てとは正数であると仮定する。)

【0012】利用不足という課題を解決するための一つの提案は、「644特許」の中で記述されているように、予備バンクまたはあふれバンクと名づけられる余分にバンク B_0 を付加することである。通常のバンクと同様に、あふれバンクもまた限定された容量を備えている。本機構は次のように動作する。(図3を参照のこと)。

- 1) クラス i に対応するトークンが契約のレート r_i で生成される。
- 2) すでに B_i の中にあるトークンの数が L_i より少ない場合は、新たに生成されたクラス i のトークンはバンク B_i の中に挿入される。そうでない場合は、トークンはあふれバンク B_0 の中に蓄積される。あふれバンクがたまたま満杯になった場合には、そのトークンは破棄される。

- 3) バンク B_i の中にトークンを発見するクラス i の到着はサービスに入るのを許可される (そして、そのトークンは破棄される)。バンク B_i の中で利用できるトークンはないが、あふれバンクが空でない場合には、クラス i 到着もまたサービスに入るのを許可される (そしてトークンがあふれバンクの中で破棄される)。そうでない場合は、クラス i の到着は拒絶されるかまたは待ち行列に入れられる。

【0013】BergerとWhittによって著述された「複数クラス入力規制スロットル」という題名の記事 (Proceedings of The 29th IEEE Conference On Decision and Control, 1990, pp.2106-2111) で、許可されない到着がシステムから消失される事例を調査している。この記事は、二つのポアソン分布ストリームの事例に対する正確な閉塞とスループットと、 $N > 2$ のクラスに対する近似モデルを論議している。(この記事はここでは参照により組み込まれている。)

【0014】すぐには許可されない到着は待ち行列に入れられ、かつクラスの数が2を越える場合は、待ち行列に入れられたジョブによるあふれバンクへのアクセス規則が規定される必要がある。

【0015】さらに、もう一つの効率的な複数クラスの入場許可制御アルゴリズムは、レート制御スロットルに類似の、リーク・パケツ機構であり、やはり「644特許」の中に記述されている。固定されたレートで流れ込むある量の素材 (たとえば、トークン) の利用可能性に注目する代わりに、リーク・パケツ機構は、固定されたレートで排出するパケツの中のスペースの利用可能性を考慮する。

【0016】 N 個の実際のパッファに加えて (待ち行列 Q_1, Q_2, \dots, Q_N)、リーク・パケツのアクセス機構は、 $N+1$ 個の仮想パッファ (リーク・パケツ $L B_0, L B_1, \dots, L B_N$) から構成される。各々のクラスは、それ自身の待ち行列と、それ自身のリーク・パケツとを割り当てられる。待ち行列の大きさは無限である一方で、リーク・パケツは有限の容量である。 $L B_i$ の容量を L_i として示す。ここで、 $i = 0, 1, \dots, N$ 。追加的リーク・パケツ ($L B_0$) は、すべてのクラスに共通である。システムに入るのを許可されるために、クラス i の到着は b_i 量の流体をそれ自身の $L B_i$ の中に、または、前者が最大量を含むことができない場合には共通の $L B_0$ に、挿入しなければならない。 $L B_i$ と $L B_0$ の中の予備の容量の和が b_i より少ない場合は、到着は Q_i の中で待つ。クラス i のトラフィックによって起動された流体は、パケツ $L B_i$ の中に、またおそれ $L B_0$ の中に連続的に挿入されて、 Q_i が空 (から) でないかぎりスペースがリークによって作られる。ライン到着のヘッドのための必要な量が挿入されるとすぐに、一つのクラス内のアクセスはFIFO規則 (すなわち、先入れ先出し方式) に従って許可される。

7

LB_i、 $i \neq 0$ は契約のレート r_i でリンクする。パケツ、たとえばLB_j、 $j \neq 0$ が空のとき、レート r_j でのその放出能力は、直ちにLB₀へ伝達される。流体がLB_jの中に挿入されるとすぐに、リンクはそれに返却される。すなわち、LB₀はそれ自身のリンクを持たない。

【0017】

【課題を解決するための手段】先行技術の不利益な点は、改善されたリソース配分のシステムおよび方法を特徴とする本発明により解決された。

【0018】一面では、本発明はリソースへのアクセスを要求する複数の使用者の間で共用され限定されたリソースにアクセスを割り当てる方法と特徴としており、この割り当ては、各使用者に関係付けられた最小限の保証されたレートでのリソースへのアクセスを各使用者が許可されるように成される。そして、リソースが利用できる場合だけ、その最小限の保証されたレートを越えている使用者によってリソースが利用できるようにしている。この方法は以下のステップから構成される。各使用者に関係付けられた最小限の保証されたレートによって決定されたレートで各使用者にトークンを割り当て、各使用者に関係付けられたバンクの中に各使用者に割り当てたトークンを蓄積して、各バンクは各使用者に関係付けられた有限の容量を備えていて、ある使用者の関係付けられたバンクがその容量に到達した場合のその使用者に割り当てたトークンを、別の使用者のバンクがその容量以下にある場合のその別の使用者に割り当てられたバンクの中に蓄積して、そして、その使用者がその関係付けられたバンクの中にあらかじめ決められた数のトークンを備えている場合には、リソースへのアクセスを要求している使用者にリソースへのアクセスを許可する。そうでない場合には、そのリソースへのアクセスを拒絶する。

【0019】好ましい実施例の中では、各使用者は異なる優先権を割り当てられて、その割り当てられた優先権に基づいて選択される。

【0020】この方法はさらに、リソースへのアクセスを許可された使用者に関係付けられたバンクから、あらかじめ決められた数のトークンを削除するステップを含む。あらかじめ決められた数のトークンは、一つのトークンであってもよい。各バンクは、あらかじめ決められた同じ有限の容量を備えているか、または少くとも二つのバンクが別の容量を備えているかのいずれかである。

【0021】もう一面では、本発明は共用されたリソースへのアクセスを割り当てる方法と特徴としており、下記のステップから構成される。

(a) リソースの複数の使用者の各々に対して加算できるカウンタを確立して、この加算できるカウンタは計数

(b) 本計数が各使用者に関係付けられたあらかじめ定

8

義された最大計数以下である場合には、各使用者に対して確立されたカウンタの計数を各使用者に関係付けられたレートで増加させて、

(c) アクセスを要求している使用者に対して確立されたカウンタの計数がゼロでない場合は、リソースへのアクセスを要求している使用者にアクセスすることを許可する。そして、

(d) ステップ(b)で増加させられる他のカウンタの中での計数がどれもそのあらかじめ定義された最大計数にある場合は、そのカウンタがあらかじめ定義された最大計数になっていない使用者に関係付けられたカウンタの中でその関係付けられたレートより高いレートでその計数を増加させる。

【0022】なお別の面では、本発明は、N個の使用者の間で共通のリソースへのアクセスを共用するための方法と特徴としており、ここでNは1より大きな整数であり、以下のステップから構成される。

【0023】N個の使用者の各々にレート「R」を割り当て、このレートは1番目の使用者に対する最小限の要求されるアクセスのレートを示しており、ここで、 $1 < N$ である。N個の使用者の各々に対して共通のリソースへアクセスするために1番目の使用者の資格のしるし「B₁」を蓄積する。N個の使用者の各々に割り当てたあらかじめ決められた限度値L₁まで各B_iをその関係付けられたレートRで増加させる。他のB_iがどれもその関係付けられた限度値L₁にあり、かつ選択されたB_iがその限度値L₁にない場合に、その選択されたB_iをその関係付けられたRより大きなレートで増加させ、そして、その関係付けられたB_iがゼロでない場合に、そのi番目の使用者に共通リソースへのアクセスを許可する。望ましくはこの方法はさらに、そのリソースの1番目の使用者にアクセスを許可する際に、B_iを削減するステップを含む。

【0024】

【発明の実施の形態】本発明の第一の実施例は、上で論議したトークン・パケツの機構の變形と特徴とする。ここでは、あるクラスは、最初にトークンを共通のバンクB₀から引き出すことができ(あるいは、LB₀にその流体の特定量を挿入)ることを許可されている)で、そして、その共通バンク(またはパケツ)が満杯であるときだけそれ自身のバンク(またはパケツ)を使用する。この機構は、システムがあるクラスに優先権を与えることを許容して、最初に使用することのできるいくつかのあふれリソースがある場合は、そのクラスが将来の使用のためにリソースへのそれ自身のアクセスを「節約」させることを許容する。

【0025】第二の実施例では、システムはあふれバンクやパケツを完全に省略することができて、その代わりに、ある種の選択された使用者に対して、その契約した容量以下で動作している使用者によって生成された超過

のトークン(またはリンク容量)を分配し直すことができる。この機構は以下に説明するが、説明を簡単にするために、リンク・パケツに類似の状況の中だけで説明される。

【0026】図4に示した機構を考察すると、共通パケツLB₀の大きさは任意に小さく作成することができ、このことは許可されたストリームのバースト(突発)時の共通パケツのインパクトが無視できることを意味している。図4は、パケツLB₂が空という特定の状況を捕らえている。したがって、その排出能力は現在、共通パケツLB₀で行使されている。また、実際の待ち行列Q₁の中で待っているクラス1の到着がある点に注意のこと。図4に示すように、クラス1の到着に対してシステムへのアクセスを許容するために要求された特定量を挿入するためのパケツLB₀とLB₁とに余地がないことをこれは意味する。(このように、クラス1の到着

$$a_n^{(i)} = \begin{cases} 1 & \text{時刻 } n \text{ での到着がクラス } i \text{ である場合} \\ 0 & \text{その他の場合} \end{cases}$$

【0028】同一時刻の到着は許されない。すなわち、20 ※含める。それから間隔Δ_{n+1}の間の排出を計算する。
a_n⁽¹⁾ a_n⁽²⁾ = 0。到着時刻の間のシステムの展開 【数2】

を記述するために、前の到着により寄与した作業負荷を※

$$\hat{z}_{n-1}^{(1)} = z_n^{(1)} + a_n^{(1)} b_n^{(1)} + \min(a_n^{(2)} b_n^{(2)}, [L_0 - z_n^{(2)}]^+) \quad (1)$$

$$\hat{z}_{n-1}^{(2)} = z_n^{(2)} + a_n^{(2)} b_n^{(2)} + \min(a_n^{(1)} b_n^{(1)}, [L_0 - z_n^{(1)}]^+) \quad (2)$$

$$z_{n+1}^{(1)} = [\hat{z}_{n+1}^{(1)} - \Delta_{n-1} r^{(1)} - [\Delta_{n-1} r^{(2)} - \hat{z}_{n-1}^{(2)} + L_0]^+]^+ \quad (3)$$

$$z_{n+1}^{(2)} = [\hat{z}_{n+1}^{(2)} - \Delta_{n-1} r^{(2)} - [\Delta_{n-1} r^{(1)} - \hat{z}_{n-1}^{(1)} + L_0]^+]^+ \quad (4)$$

【0029】方程式1-4は共通パケツLB₀を持つことのインパクトを強調する。特定のパケツ、たとえばLB₁の展開を検討する。方程式3の中のマイナスの項はLB₂による共通の排出への寄与を表す。Δ_{n+1} r⁽²⁾、つまりその間隔の間のその排水能力が、それ自身の排他的な作業負荷z_{n+1}⁽²⁾ - L₀を越えるとき、後者は共通の排出に寄与する。この共通の排出能力が、L₀を持つことのきわだった特徴である。一方、方程式3の右★

$$Pr(L_0^{(1)} > x) = Pr(L_0^{(2)} > L_0 + L_1 + x) \quad (5)$$

方程式1と2とは結合される。すなわち、z_n⁽¹⁾、i = 1, 2のCDFを得ることは、(z_n⁽¹⁾ z_n⁽²⁾)の結合分布を見つけることを必要とする。状態空間が、3というよりはむしろ2であるということに注意のこと、我々が共通のパケツと自分自身のパケツに挿入する際に順序を逆にする場合にそうなる。このことは、D_n⁽¹⁾、i = 1, 2のCDFを計算するための数式手順の中で相当な簡略化を表す。

【0030】それを共通パケツなしの複数クラスのアクセスと比較することによって、アクセス方式に対するか

*着はQ₁で待っている。)

【0027】補助変数Z_n⁽ⁱ⁾、i = 1, 2で、その到着をカウントしない到着時刻nでの、LB₀、LB₁、Q₁の中の作業負荷の総和を示すこととする。また、その待ち行列とリンク・パケツの中の作業量として、Q_i、LB₁、LB₀を示す。図4に示した特定の状態の中で、Z₁ = LB₀ + LB₁ + Q₁、Z_n⁽²⁾ = LB₀。流体の特定量b_iの挿入が完了するまで到着はQ_iの中で待たなくてはならない。しかし流体は、Q_i > 0、LB₁ + LB₀ - L₁ - L₀ < 0であるかぎり、パケツからリンクされる。また、LB₁ > 0 > LB₀ = L₀に注意のこと。システムへの第n番目の到着の時刻を示すのに索引nを使用し、時刻nと時刻n+1の間の時間をΔ_{n+1}で示し、確率変数を導入する。

【数1】

30 ※辺の第三項によって示されるように、z_{n+1}⁽¹⁾はクラス2の到着によっていくぶん影響を与えられる。しかし、この項が示すように、インパクトはL₀を越えることができないために、我々の関心は、到着時刻nでのQ_i、i = 1, 2の中での作業負荷D_n⁽ⁱ⁾の累積分散関数(CDF)を見つけることである。それゆえに、

【数3】

40 なる洞察を得ることができる。上記したように、そのような機構は保護の観点から効果的だが、効率に欠けている。ここに、二つのクラスという特定の事例に対するポイントを証明する。この目的のために、我々のオリジナルのシステムと同じ到着に関わる共通パケツのない別のアクセス機構を検討して、その変数に差異を付けるためにバーを使用する。特に、方程式1と2とは、次式に縮小できて、

【数4】

$$\mathbb{Z}_{n+1}^{(i)} = [\mathbb{Z}_n^{(i)} + a_n^{(i)} b_n^{(i)} - \Delta_{n+1} r^{(i)}]^+, i=1, 2, \quad (6)$$

そして、方程式5を下記に変換する。

$$Pr(\mathbb{Z}_n^{(i)} > x) = Pr(\mathbb{Z}_n^{(i)} > L_i + x), \quad \text{【数5】}$$

(7)

【0031】

【外1】

$\bar{L}_i = L_i, i=1, 2, L_0 = e$ を選び、そして、共通パケツの大きさへの作

業負荷の依存を強調するために、 $\mathbb{Z}_n^{(i)}(e)$ を書く。

$$W_n^{(i)} = \lim_{e \rightarrow 0} \mathbb{Z}_n^{(i)}(e) \quad \text{として、} \quad W_0^{(i)} = \mathbb{Z}_0^{(i)}, i=1, 2 \text{ とおく。}$$

すると、任意の整数 $n > 0$ に対して、確率変数 $W_n^{(i)}$ は $\mathbb{Z}_n^{(i)}$ より確立論的に小

さい。すなわち、すべての $X \geq 0$ に対して、 $Pr(W_n^{(i)} > x) \leq Pr(\mathbb{Z}_n^{(i)} > x)$

【0032】 $L_0 \rightarrow 0$ になると、クラス i (方程式3と * ラスは、 $\mathbb{Z}_n^{(i)}$ の排出に寄与する (方程式3と4)。
4) の作業負荷への他のクラスの寄与がゼロに減少する 【外2】

点に注意のこと。しかし、 $L_0 = 0$ に対しても、他のク*

したがって、限度値の中で、 $\mathbb{Z}_n^{(i)}(e)$ の展開は、正でない項の加算で $\mathbb{Z}_n^{(i)}$

の展開と一致する。

【0033】 以下の推論は、共通パケツを備える複数ク
ラスのアクセス制御装置はアクセス遅延という見地から
古典的なアクセス制御装置より性能が優れていることを※

※示す。

【0034】

【外3】

上述した条件のもとで、確率変数 $W_n^{(i)}$ は、すべての $X \geq 0$ に対して $W_n^{(i)}$

より小さい。すなわち、 $Pr(W_n^{(i)} > x) \leq Pr(\mathbb{Z}_n^{(i)} > x)$

この不等式は再度書き直すことができ、

【数6】

$$Pr(W_n^{(i)} > L_i + x) \leq Pr(\mathbb{Z}_n^{(i)} > L_i + x)$$

これは上記の提案に従ったままである。

【0035】 上記の注記のように、使用者が余分なパケツ $L B_0$ を最初に調査して、 $L B_0$ が満杯である場合にはそれ自身のパケツの中に流体を挿入するという、古典的なリーク・パケツ機構への変更が提案された。それから、提案された機構の動作は標準の多重アクセスのアプローチと比較された。一貫した比較を行うために、パケツ $L B_0$ の大きさはゼロに設定された。そうすることのプロセスの中で、余分な容量を分配し直すための機構は、主としてリーク能力の再配布に依存しており、余分なパケツ $L B_0$ には依存しないことが明確になった。それゆえに、余分なパケツは全く省略でき、アルゴリズムは以下のように修正される。パケツ $L B_1$ が空であるときはいつでも、レート r_1 でのその排出するための能力は、空でないパケツの間で分配される。リークを再分配するために二つのアルゴリズムを使用することができ。

30 1) 契約のレートに比例した超過容量の再分配。 $L B_1$ が空だとする。 $A = \{L B_i : L B_i > 0\}$ とおく。次に、パケツ $L B_i$ のリークのレート $i \in A$ を

$$r_i = \frac{r_1}{\sum_{j \in A} r_j}.$$

だけ増加させる。

2) 優先権テーブルに従った超過容量の再分配。削減する優先権順にパケツを順序づける。使用されていないリークは、最初に優先度リストの先頭になる。このパケツが空であるときだけ、それはリストの第二番目へ移動する。以下同様である。

【0036】 この二つの機構が、公正さの特定の解釈から差別的な配置へと広範囲にわたるレンジの阿極端である。他の可能性は、最も要求の高いパケツに超過分を割り当てることが含まれる。

【0037】 本発明のこの実施例に対して少なくとも二つの注目すべき効果がある。第一に、レートはシステムへの平均到着率に関係するが、パケツの大きさはシステムが耐えることのできるバースト (突発性) に関係付けら

13

れる。その最悪バーストが、標準のリンク・パケツをクラスあたりに割り当てる場合のバーストに等しいという機構を備えていることは有利である。第二に、超過容量の分配に焦点をしばった、より巧みな実現が達成される。

【0038】システムの故障、あるいはスロットルによって制御されない外部トラフィックは、なお時折システム過負荷のシナリオに陥ることがある。レート制御スロットルは開閉ループ制御装置である一方で、レート $\{r_i\}$ のセットをシステムの状態に従属させることによって、遅い時間・規模の順応ループを提供することが可能となる。

【0039】上に提案された余分なパケツのないアクセス制御装置は、余分な容量を分配するためにレートの組込み式上方向応機構を備えていて、閉じたループを持つ過負荷時での遅い時間・規模の下方向応とは対照的である。実際に、制御されていないトラフィックに起因する混雑のために全体のレートが減少される一方で、不活動な使用者の寄与のためにいくつかのクラスに關保付けられた現実のレートが増加されることは、確かに可能である。

【0040】以前のアクセス制御機構は、一群のリソースへの複数のレベルでのアクセスを制御するために使用することができる。たとえば、一つのシステムを使用して多数の顧客の間のサービス（たとえば無線）へのアクセスを制御して、それからもう一つの機構を使用してリソースに対する多数の異なるサービスの間でアクセスを制御するのは望ましいことである。図9はそのようなシステムを示しており、ここでは多数のセグメント $a-z$ がサービス1-Nの中の一つへのアクセスのために優先権セグメント・アクセス制御装置（PSAC）20の制御の下に最初に競合する。その後、複数サービス・アクセス制御装置（MSAC）30の制御の下に、サービスはリソース40へのアクセスを争う。MSAC30は、望ましくは上で陰線されたアクセス制御機構の一つを使用する。（便宜上、「トークン」類似を使用して記述したが、リンク・パケツ類似もまた使用してもよい）。

【0041】各々のPSAC20は以下の好ましい方法に従ってその対応するサービスへのアクセスを制御する。図6を参照すると、多数のバンク $B_1 - B_N$ が示されており、各々はシステムの利用者のNクラスの一つに対応する（各クラスは使用者セグメント $a-z$ に対応している）。各クラスのバンク B_i はトークンのストリームを契約のレート r_i で受け取る。また、あふれバンク O_i は各クラスと關保付けられている。特定のクラスに対して生成されたトークンが満杯のバンク B_i を発見した場合、そのトークンはそのクラスのあふれバンク O_i に直ちに挿入されない。その代わりに、システムは最初に余分なトークンをバンク O_i に挿入しようとする。バンク O_i はクラス i のあふれバンクを示し、このシ

14

テムの中のより高い優先順位クラスである。あふれバンク O_i が満杯である場合には、該トークンはあふれバンク O_2 に挿入される。以下同様に続く。

【0042】このように、各クラスがその契約のレートを保証される一方で、超過容量はあるクラスに味方する優先権機構に分配される。

【0043】要求された優先権順位機構が実行されるのに適切にように、この機構を変更することができる。たとえば、単一の高優先順位クラスは、あふれトークンをこのクラスのあふれバンクに最初に挿入しようとしている各々のクラスに指名することができる。そしてこのあふれバンクが満杯の場合には、それ自身のあふれバンクの中に超過トークンを挿入する。あふれバンクの大きさはまた、システムの好ましい優先権を反映するために変更することができる。

【0044】プロセッサ共用のための複数クラスのポーリング・アルゴリズムをこれから記述する。N個の利用者によって共用されるプロセッサについて考察する。プロセッサのリアルタイムの部分 $f^{(0)}$ がオーバーヘッドに割り当てられるとする。（典型的に $f^{(0)} \approx 0.1 \sim 0.15$ ）オーバーヘッドは、総計 $N+1$ 個の利用者に対して 使用者0 と見なされる。

【0045】使用者1 によって契約されたリアルタイム使用の部分 $f^{(1)}$ で示すとする。 $\sum_{i=0}^N f^{(i)} = 1$ と仮定する。契約した部分と要求されたオーバーヘッド $f^{(0)}$ を加えた総和が1より少ない事例については、正規化に注意が必要である。使用者1 のジョブ要求は、バッファ B_1 の中に蓄積される。バッファ B_0 は、オーバーヘッド作業のために予約される。時間は、持続時間 τ のスロットに分割される。細分性はプロセッサに依存しており、そのトレードオフは識別するのが簡単である。細分性が粗すぎるのは制御がしにくくなり、細分性を微細にしすぎると容認できないオーバーヘッドをもたらすことがある。

【0046】 γ セットにプロセッサが、バッファ B_i のセット $B = \{B_1, 0 \leq i \leq N\}$ の一つから γ 秒に値する作業を拾い上げなければならない。プロセッサは下記のような方法で予定されている。

- a) その契約の限度内で動作するあらゆる使用者はそれが期待するサービスを受け取ることと保証する。
- b) その契約の限度を越えて動作するあらゆる使用者は少なくともその契約上の得点を受け取ることと保証する。そして、
- c) 作業がいずれかのバッファにある場合は、プロセッサはどの巡回の中でもサービスを拒否しない。サービス規律は、プロセッサが、 γ 秒以下しか値しない作業でバッファをサービスする点において本当は作業を保護しないことに注意のこと。また、 $\gamma \rightarrow 0$ にすると、サービス規律は作業を保護するようになる点にも注意のこと。

【0047】時刻 n で、制御変数 $U_n^{(1)}$ は、バッファ B_i に関する決定を示す。 * 【数8】

$$u_n^{(1)} = \begin{cases} 1 & B_i \text{ が選択された場合} \\ 0 & \text{そうでない場合} \end{cases} \quad (8)$$

どのバッファ (B_0 を含む) も空にならないシナリオを、※回数 i を示すことに注意のこと。契約の部分 ($i^{(1)}$) を考慮する。 $\sum_{j=1}^n U_j^{(1)}$ は、プロセッサが、時刻 n を含めて n まで、バッファ B_i からセグメントを選ん※ 【数9】

$$\lim_{n \rightarrow \infty} \frac{\sum_{j=1}^n u_j^{(1)}}{n} \geq x^{(1)}, i=0, 1, \dots, N. \quad (9)$$

【0048】この長期にわたる平均化手段は必要だが、★実際の配置時刻の最大偏差を最小にしようとする。【数10】
使用者の期待に適合するには十分ではない。それから、下記によって測定されるような目標値から使用者 i への★

$$e_n^{(1)} = \sum_{j=1}^n [x^{(1)} - u_j^{(1)}], \quad (10)$$

これは帰納的にアップデートすることができ、 【数11】

$$e_0^{(1)} = 0, \quad (11)$$

$$e_{n+1}^{(1)} = e_n^{(1)} + [x^{(1)} - u_{n+1}^{(1)}], n=0, 1, \dots \quad (12)$$

【0049】時刻 n でサービスされたバッファを b_n * ☆1) 初期状態設定：
で示して、以下のアルゴリズムを検討すると、 ☆ 【数12】

$$e_0^{(1)} = 0, \quad i=0, 1, \dots, N, \quad (13)$$

$$n = 0.$$

2) ボーリング時

【外4】

a) ボーリング決定:

$$b_n = \operatorname{argmax}_i e_n^{(1)} \quad (14)$$

(辞書式順序との連結を切る)

$$u_n^{(1)} = \begin{cases} 1 & b_n = 1 \text{ の場合} \\ 0 & \text{そうでない場合} \end{cases}$$

b) 状態アップデート:

$$e_n^{(1)} = e_{n-1}^{(1)} + [x^{(1)} - u_n^{(1)}], \quad (15)$$

【0050】したがって、好ましい実施例の中で、時刻 n でのボーリング決定は時刻 $n-1$ で最も大きな $e^{(1)}$ を選択することによって成される。このように、プロセッサへのアクセスは、その契約した持ち分とその実際の使用オーバー時間の間の最も大きな差を示す使用者に許可される。これは、すべての使用者がプロセッサに連続的にアクセスを要求しているとき (すなわち、使用者のバッファが決して空にならない時) の、プロセッサ共有の最も公正な方法である。

【0051】しかし、最も現実的なシナリオでは、サーバの利用は1以下でありバッファが空になる。また、ジョブは通常、セグメントに分割される。セグメントの長

さはCPUがタスクを実行するために取る時間を示し、したがって、すべてのセグメントが同じ持続時間を持つと仮定することは非現実的である。次に、これらの現実的な課題を考慮した別の実施例を検討する。

【0052】 $S_n^{(1)}$ をボーリング時刻 n でのバッファ B_1 のヘッドでのセグメントの長さとして、そのセグメントの長さに対してある種の上限 S_{max} があると仮定する。二つの特定の事例、すなわち異質の過負荷と同質の通常の負荷のシナリオに焦点をあてるのが実証となる。過負荷で異質の事例は、方程式15の単純な変更によって、簡単に扱うことができる。

【数13】

$$e_n^{(i)} = e_{n-1}^{(i)} + S_n^{(b_n)} [f^{(i)} - u_n^{(i)}]$$

【0053】すなわち、変更は、ボーリングされたバッファの先頭で、セグメントをサービスしているプロセッサによって費やされる時間によって、クレジット（貸方） $f^{(i)}$ と装置借方 $u_n^{(i)}$ の両者を重み付けすることである。

【0054】同質の通常負荷の事例を考慮すると、各々のクラス i がCPUの同じ部分、 $f^{(i)} = 1/N+1$ 、 $i = 0, 1, \dots, N$ を割り当てられると仮定する。セグメントの長さの分布は、すべてのクラスに対して同じである。このシナリオの中で、自然で、公正で、かつ最も合理的な基準に従う本当に最適な方針は、ラウンド・ロビン・アルゴリズムであり、そこではサーバは空のバッファをスキップする。

【0055】ボーリング・バッファが空であるとき、システムは下記のように処理する。プロセッサがセグメントの到着を待つことは無駄であるので、空のバッファはスキップされると決定する。それから、問題は空のバッファへの「仮想」の巡回をどのように説明するかということである。仮想巡回に対して空のバッファを満たさないことを考慮してもよく、それに経過時間に対してクレジットを与え続けてもよい。このアプローチの持つ課題*

a) ボーリング決定

$$b_n = \underset{i}{\operatorname{argmax}} e_{n-1}^{(i)}$$

（辞書式順序との連結を切る）

$$u_n^{(i)} = \begin{cases} 1 & b_n = i \text{ の場合} \\ 0 & \text{そうでない場合} \end{cases}$$

（ i は仮想セグメントであり、ここではプロセッサは時間を費やさない。）

b) 状態アップデート

$$e_n^{(i)} = e_{n-1}^{(i)} + S_n^{(b_n)} [f^{(i)} - u_n^{(i)}].$$

【0057】上述したように、たとえリソースへのアクセスが統制されるとしても、それにもかかわらず一つ以上の特定の個別のリソースへのアクセスを統制することは望ましいことである。個別のリソースにアクセスするものの複数クラスの巡回トークン機構について次に説明する。

【0058】トランク、3者会合回線、UTD（汎用トラン・デコーディング）は、個別のリソースの中にあり、異なるクラスの使用者の間で共有される必要がある。そのま

18

*題は、不活動な発信源によって累積されたクレジットが、それが目覚めたとき、CPUを独占することがありうるということである。合理的な解決策は帰納的に指数的な平滑化を導入することであり、それによって過去よりも現在に多くの重み付けがなされる。もう一つの選択肢は、活動的なバッファの中だけで各段階でクレジットを配布することである。別のアプローチは、選択されたバッファが仮想回線に対する「失われた機会」を負担させられるという概念に基づいている。

【0056】今や、一般的なプロセスを述べるができる。 $S_n^{(i)}$ がボーリング時刻 n でのバッファ B_i の先頭でのセグメントの長さであることを思い出そう。バッファ B_i が空であるとする、これを S_i にセットする。ここで、 S_i つまり使用者 i の平均のセグメント長は下記のとおりであり、これを仮想セグメントと呼ぶ、

1) 初期状態設定：

【数14】

$$e_0^{(i)} = 0 \quad i=0, 1, \dots, N, \\ n = 0.$$

2) ボーリング時

【外5】

さしくその本質によって、これらのリソースは、ビジーであるかまたは動作していない。通常、リソースの保持時間は、その状態を伝送するために必要とされる時間をはるかに越える。そしてもちろん、リソースの個別の性質は正の整数に写像される。以下は、共通のプールをもつ巡回トークン機構である。

【0059】 N 個の使用者が競合する M 個の装置のリソースがあるとする。

仮定、

1. n_i のトークンをクラス i のバンクに割り当てる。

$i = 1, 2, \dots, N$

2. n_0 のトークンを共通のバンク b_0 に割り当てる。
したがって、

【数15】

$$n_0 + \sum_{i=1}^N n_i = M$$

【0060】各々のトークンがリソースの一つの装置にアクセスを与える。トークンは巡回されるが、破壊されない。すなわち、アクセスを許可した装置がフリーになると、取られたバンク（自分自身のバンクまたは共通バンク）にトークンは返却される。

【0061】たとえば、クラス i が $m \geq 1$ 個の装置のリソースを必要すると仮定しよう。3つの可能性がある。

(1) それぞれが自分の m 個の不動作中のトークンを持っていて、この場合、トークンと引き換えに要求する装置へのアクセスを得る。(2) それはそれ自身の m 個以下のトークンしか持っていないが、共通バンクの中

のトークンは要求を完了するのに十分であり、クラス i が、(1) のように処理する。そして、(3) それ自身の利用できるトークンと共通バンクの利用できるトークンとの和が m 個以下の場合には、システム特有の設計によって、その要求が拒絶されたり待ち合わせせられたりする。

【0062】この機構は、リソースの完全な分割から全面的な共用まで、広範囲にわたる解決策をカバーする。トークンが共通バンク ($n_0 = 0$) に一つも割り当てられないとき、完全なリソース分割に変わる。他の極端では、 $n_0 = N$ (したがって、 $n_i = 0, i = 0$) の場合、完全にリソースを共用する。完全な分割は効果的である。クラス i は常に、 n_i 個の装置のリソースへの同時アクセスを保証される。残っている $N - n_i$ 個のすべての装置が動作していない場合でも、 n_i 個以上のアクセスを全く得ることができないことは効率的でない。一方、完全な共用では、この最後のシナリオを見事に処理するが、 n_i 個の装置へのあらゆる時刻でのクラス i のアクセスを保証しない。中間の状況 $0 < n_0 < N$ は、保証と柔軟性との間でのトレードオフを表す。共通のプールに寄与するために、クラス i は $n_i < n_i$ に配置されることになる。同時に使用することのできる装置の最大数はシステムに集合された負荷によって、 n_i と $n_i + n_0$ との間で変化する。

【0063】機構の変更の一つは、トークンをそれ自身のバンクの中で探索する前に使用者のサブセット（高い優先順位クラス）に共通バンクへのアクセスを与えることから成る。正確な応答を得るために、トークンを返却するための規則の変更が必要とされる。この変更に従うと、トークンは常に優先権の高いバンクに返却されて、そこが共通バンクにあふれ出る。これらの変更は、一般にトークンの巡回方針が下記によって決定されることを

示す。

- トークンとトークン・バンクとを異なるクラスへ割り当てる。
- 一つまたは二つ以上の共有されたバンクにトークンを割り当てる。
- バンクへのアクセスを許可する規則。
- 使用されたトークンをどのバンクへ返却しなければならないかを決定する規則。

【0064】複数のクラスのレートのアクセス制御と巡回トークン機構の間にはいくらかの類似性がある。両方のアプローチは、トークンの概念、および共通の装置と同様にクラスに特有の装置の概念という両方の観点から説明される。しかし、それらが向かう課題の間の差異を明確にする重要な違いがある。トークンを巡回させる閉ループ特性は、複数のクラスのアクセス制御の開ループ特性と対照をなす。また、前者のトークンは本来的に個別的であるが、後者は実数値のトークンを認める。トークン巡回機構の属性は、個別的なリソースの明かな本質から生じる。装置は使用中かまたは不動作中であり、しかもこの状態は多くの場合リアルタイムで利用できる。一方、図1で描写されたようなグローバルなシステムに入るのを許可される到着は、より巧妙な方法で多数の異なるリソースを消費するように連結される。これは、ある種のプロセッサの利用法を増加させ、ある種の待ち行列の中で遅延を増大させる。また、これらの内部リソース（プロセッサ、リンク、その他）のインバントは、到着と関係付けられた活動の保持時間の間の異なるポイントで発生する。

【0065】以上の説明は、事例としてだけを意図した好ましい実施例の図を含んでおり、添付の請求項の適用範囲内で種々の他の実施例が可能である。

【図面の簡単な説明】

【図1】アクセスを求めている複数のクラスを持つAT & Tの5ESS[®]交換機のプロック図である。

【図2】先行技術のスロットル制御技法を示す図である。

【図3】第二の先行技術のスロットル制御技法を示すブロック図である。

【図4】第三の先行技術のスロットル制御技法を示すブロック図である。

【図5】本発明のアクセス制御技法を示すブロック図である。

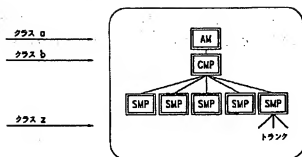
【図6】多数のセグメント間のサービスへのアクセスと多数のサービスの間のリソースへのアクセスとの両者を制御するための、本発明に従ったシステムを示すブロック図である。

【符号の説明】

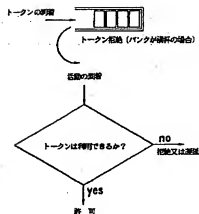
20 PSAC（優先セグメント・アクセス制御装置）

30 MSAC（複数サービス・アクセス制御装置）

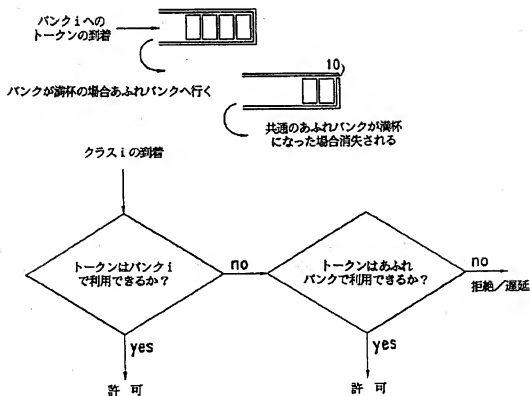
【図1】



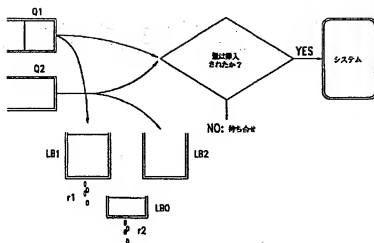
【図2】



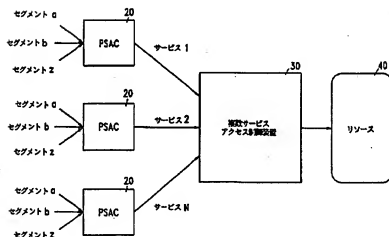
【図3】



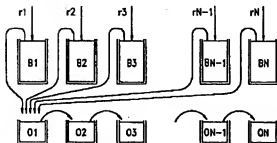
【図4】



【図5】



【図6】



フロントページの続き

(51) Int. Cl.⁶

H04Q 3/545

識別記号

庁内整理番号

9744-5K

9744-5K

FI

H04L 11/20

技術表示箇所

G

I02C

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representation of
The original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.